

# Android™ Quick Start Guide

## 1 Overview

This document guides you through the processes of downloading and running this release package. It only explains how to download and run the default release image with default configuration. For details on using the release package, see the *Android™ User's Guide* (AUG) included in this release package.

## 2 Hardware Requirements

The hardware requirements for using this release package are as follows:

Supported system-on-chips (SoCs):

- i.MX 8M Mini
- i.MX 8M Quad

Supported boards:

- i.MX 8M Mini EVK Board and Platform
- i.MX 8M Quad EVK Board and Platform

### Contents

1	Overview.....	1
2	Hardware Requirements.....	1
3	Working with the i.MX 8M Mini EVK Board.....	2
4	Working with the i.MX 8M Quad EVK Board.....	8
5	Revision History.....	16



## 3 Working with the i.MX 8M Mini EVK Board

### 3.1 Board hardware

The figures below show the different components of the i.MX 8M Mini EVK board.

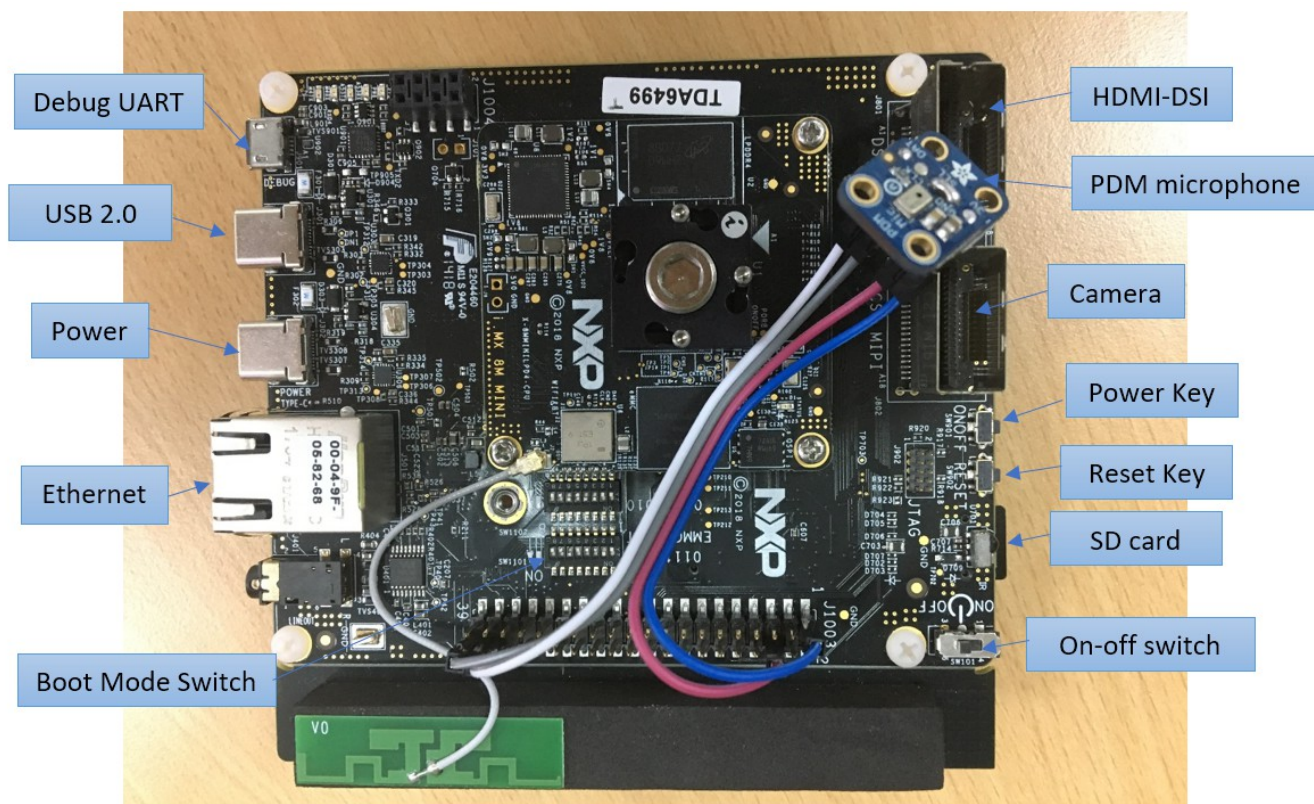


Figure 1. i.MX 8M Mini EVK board

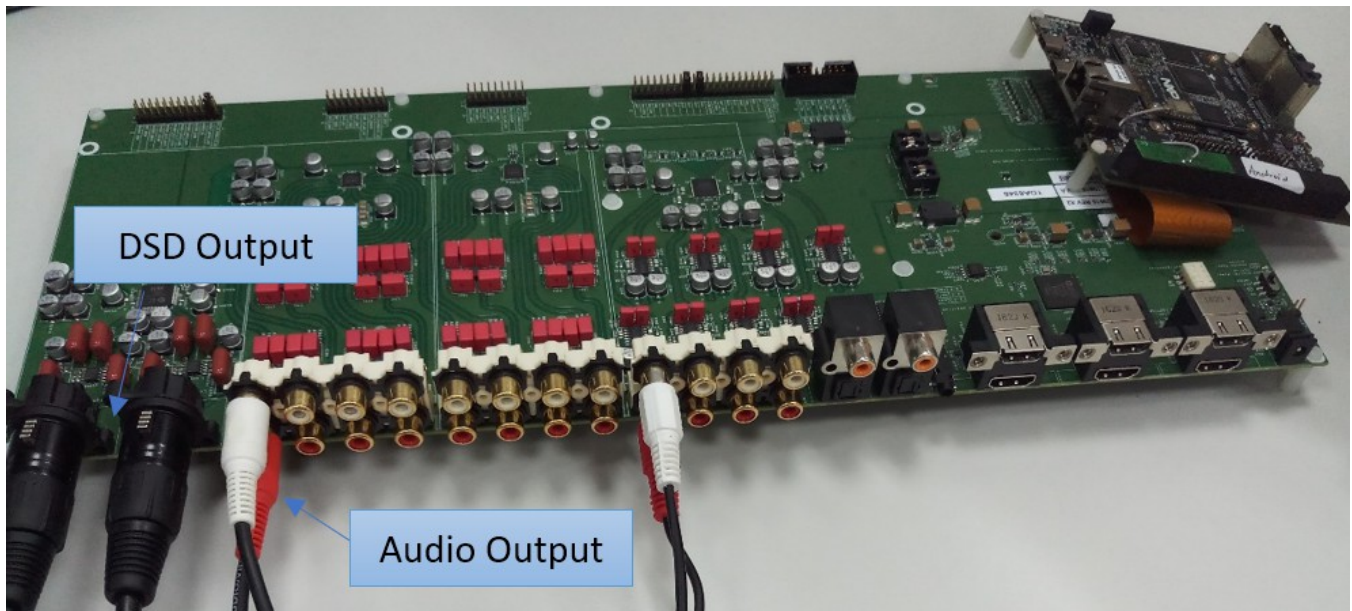


Figure 2. i.MX 8M Mini EVK with audio board



Figure 3. i.MX 8M Mini SAS cable with DSI-to-HDMI adapter





Figure 4. i.MX MIPI panel

## 3.2 Board images

The table below describes the location in the board partitions of the software images in `android_o8.1.0_1.5.1_8mm-beta_image_8mmevk.tar.gz`.

Table 1. Board images

Image name	Download target
/u-boot-imx8mm.imx	33 KB offset of MMC.
/imx8mm_m4_demo.img	5120 KB offset of MMC.
/partition-table.img	0 offset of MMC. If the actually size of the SD card is larger than 13 GB, use the default partition-table.img
/partition-table-7GB.img	0 offset of MMC. If the actually size of the SD card is larger than 7 GB, use this image as partition-table.img.
/partition-table-28GB.img	0 offset of MMC. If the actually size of the SD card is larger than 28 GB, use this image as partition-table.img.
/boot-imx8mm.img	boot_a and boot_b partitions to support MIPI-DSI to HDMI output.
/boot-imx8mm-dsd.img	boot_a and boot_b partitions to support MIPI-DSI to HDMI output and Direct Stream Digital (DSD) playback.

Table continues on the next page...

**Table 1. Board images (continued)**

/boot-imx8mm-m4.img	boot_a and boot_b partitions to support MIPI-DSI to HDMI output and audio playback based on Cortex-M4 FreeRTOS.
/boot-imx8mm-mipi-panel.img	boot_a and boot_b partitions to support MIPI panel output.
/vbmeta-imx8mm.img	vbmeta_a and vbmeta_b partitions to support MIPI-DSI to HDMI output.
/vbmeta-imx8mm-dsd.img	vbmeta_a and vbmeta_b partitions to support MIPI-DSI to HDMI output and DSD playback.
/vbmeta-imx8mm-m4.img	vbmeta_a and vbmeta_b partitions to support MIPI-DSI to HDMI output and audio playback based on Cortex-M4 FreeRTOS.
/vbmeta-imx8mm-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support MIPI panel output.
/system.img	system_a and system_b partitions.
/vendor.img	vendor_a and vendor_b partitions.

The table below describes the Universal Update Utility (UUU) scripts in android\_o8.1.0\_1.5.1\_8mm-beta\_image\_8mmevk.tar.gz. They are used with the UUU binary file to download the images above into the board. For detailed information on how to download images with UUU, see Section "3.3 Flashing board images".

**Table 2. UUU scripts**

UUU script name	Function
uuu-android-mx8mm-evk-emmc.lst	Used with the UUU binary file to download image files into eMMC. The m4_os partition is not flashed.
uuu-android-mx8mm-evk-sd.lst	Used with the UUU binary file to download image files into the SD card. The m4_os partition is not flashed.
uuu-android-mx8mm-evk-emmc-m4.lst	Used with the UUU binary file to download image files into eMMC. The m4_os partition is flashed.
uuu-android-mx8mm-evk-sd-m4.lst	Used with the UUU binary file to download image files into the SD card. The m4_os partition is flashed.

### 3.3 Flashing board images

The board image files can be flashed into the target board using UUU.

For the UUU binary file, download it from github: [uuu release page on github](#). You can download the latest version.

- For Linux OS, download the file named "uuu".
- For Windows OS, download the file named "uuu.exe" and "libusb-1.0.dll", which need to be in the same directory.

For detailed information on UUU scripts, see Section 3.2 "[Board images](#)".

#### NOTE

UUU uses the fastboot tool to flash images. Make sure you have fastboot driver software installed on your computer.

Perform the following steps to flash the board images:

1. Download the UUU binary file from github as described above.
2. Change the first two bits of the board's sw1101 to 10 (1-2 bit) to enter serial download mode.

## Working with the i.MX 8M Mini EVK Board

3. Power on the board. Connect the PC with the board using the USB cable on the board USB 3.0 port.
4. Decompress release\_package/android\_o8.1.0\_1.5.1\_8mm-beta\_image\_8mmevk.tar.gz, which contains the image files and UUU scripts. Choose the correct UUU script file as shown in the following table.

Target device and boot storage	UUU script file
i.MX 8M Mini EVK eMMC	uuu-android-mx8mm-evk-emmc.lst
i.MX 8M Mini EVK SD	uuu-android-mx8mm-evk-sd.lst

To Test MIPI-DSI to HDMI output and audio playback based on Cortex-M4 FreeRTOS, choose UUU script as shown in the following table.

Target device and boot storage	UUU script file
i.MX 8M Mini EVK eMMC	uuu-android-mx8mm-evk-emmc-m4.lst
i.MX 8M Mini EVK SD	uuu-android-mx8mm-evk-sd-m4.lst

### NOTE

- If your SD card is 16 GB or the on-board eMMC is used as the boot device, to test the MIPI-DSI to HDMI output, you do not need to change the UUU script.
- If your SD card is 32 GB, rename partition-table.img to partition-table-28GB.img in the corresponding UUU script.
- If your SD card is 8 GB, rename partition-table.img to partition-table-7GB.img in the corresponding UUU script.
- To test the MIPI-DSI to HDMI output and Direct Stream Digital (DSD) playback, rename boot-imx8mm.img and vbmeta-imx8mm.img to boot-imx8mm-dsd.img and vbmeta-imx8mm-dsd.img in the corresponding UUU script.
- To test the MIPI panel output, rename boot-imx8mm.img and vbmeta-imx8mm.img to boot-imx8mm-mipi-panel.img and vbmeta-imx8mm-mipi-panel.img in the corresponding UUU script.

5. Use UUU and proper script file to flash image files.

Execute the following command to invoke the UUU binary file and UUU scripts to flash the image files.

- On the Linux system, open the shell terminal, and change the working directory to the directory that contains the UUU binary file. Execute the command below. `{uuu_script_path}` is the file path (including the name of the UUU script) of the UUU script that is used. It can be a relative path or an absolute path.

```
> sudo ./uuu {uuu_script_path}
```

- On the Windows system, open the cmd interface, and change the working directory to the directory that contains the UUU binary file and the DLL file. Execute the command below. `{uuu_script_path}` is the absolute file path (including the name of the UUU scripts) of the UUU script.

```
> uuu.exe {uuu_script_path}
```

6. Wait for the script file execution to be completed. If there is not any error, you will get information on the command window as follows:

```
C:\Users\user_01\tools\uuu>uuu.exe C:\Users\user_01\images
\images_to_flash_and_uuu_scripts\uuu-android-mx8mm-evk-emmc.lst
uuu (Universal Update Utility) for nxp imx chips -- libuuu-1.1.30-g9f1b007
```

```
Success 1      Failure 0
```

```
2:2      15/15      [Done                ] FB: done
```

7. Change the boot device as eMMC or SD card.

- Change sw1101 to 01110010 and change sw1102 to 00101010 if you want to boot from eMMC.
- Change sw1101 to 01000110 and change sw1102 to 00110100 if you want to boot from SD card.

Problems may be encountered when using UUU:

- UUU uses the fastboot tool to flash image files. Therefore, if there are Android images in the target storage device, the device should be unlocked first. For how to unlock the device, see Section 6.2.3 "DM-verity configuration" in the *Android™ User's Guide* (AUG). If you cannot unlock the device, see Section 5.1.2 "Downloading images with MfgTool/UUU" in the *Android™ User's Guide* (AUG) to skip the unlocking operation, but this will take longer time to flash the images.

## 3.4 Booting

After downloading the images, reboot the board using the power on/off switch.

### 3.4.1 Booting with MIPI-to-HDMI or MIPI panel display

In the U-Boot prompt, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200 init=/
init androidboot.console=ttyMXC1 consoleblank=0 androidboot.hardware=freescale cma=800M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

### 3.4.2 Booting with MIPI-DSI to HDMI display and audio playback based on Cortex-M4 FreeRTOS

In the U-Boot prompt, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200 init=/
init androidboot.console=ttyMXC1 consoleblank=0 androidboot.hardware=freescale cma=800M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never
U-Boot > setenv bootcmd "bootmcu && boota mmc0" # for SD boot
U-Boot > setenv bootcmd "bootmcu && boota mmc1" # for emmc boot
U-Boot > saveenv
```

#### NOTE

To use other boot images, do not add "bootmcu" to "bootcmd". The following command can recover bootcmd:

```
U-Boot > setenv bootcmd "boota mmc0" # for SD boot
U-Boot > setenv bootcmd "boota mmc1" # for emmc boot
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

## Working with the i.MX 8M Quad EVK Board

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive  
U-Boot > saveenv
```

### NOTE

Boot image supports DSD playback (boot-imx8mm-dsd.img) and uses Single HDMI Display. They share the same U-Boot environment variables.

## 3.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

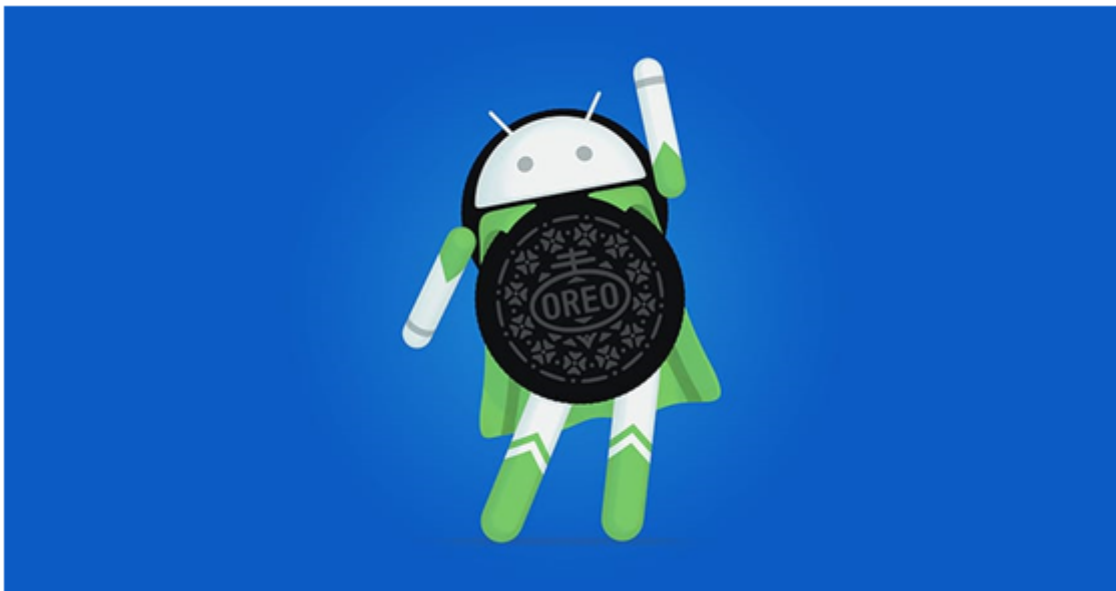


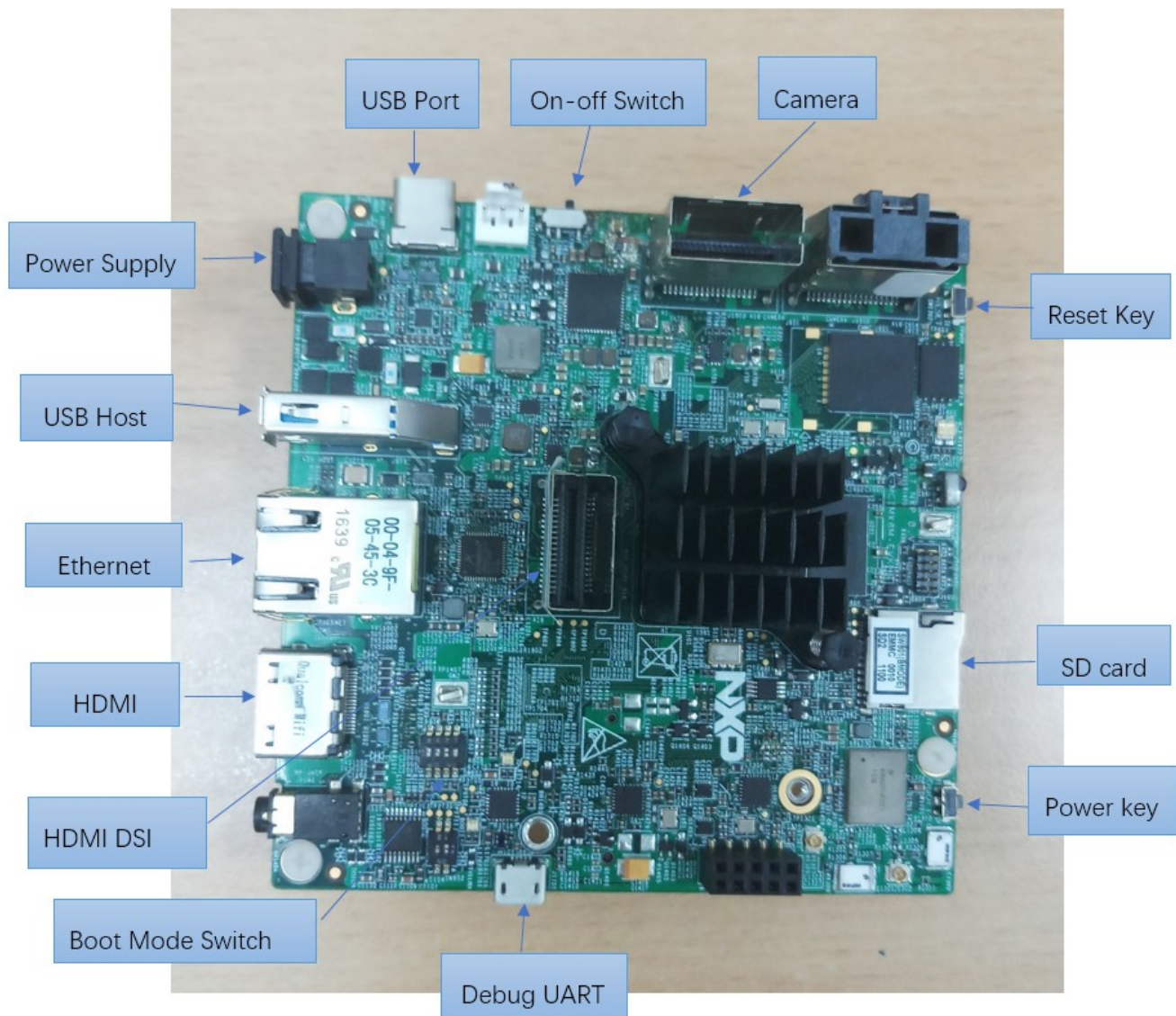
Figure 5. Android Oreo image

## 4 Working with the i.MX 8M Quad EVK Board

### 4.1 Board hardware

The figures below show the different components of the i.MX 8M Quad EVK board.





**Figure 6. i.MX 8M Quad EVK board**

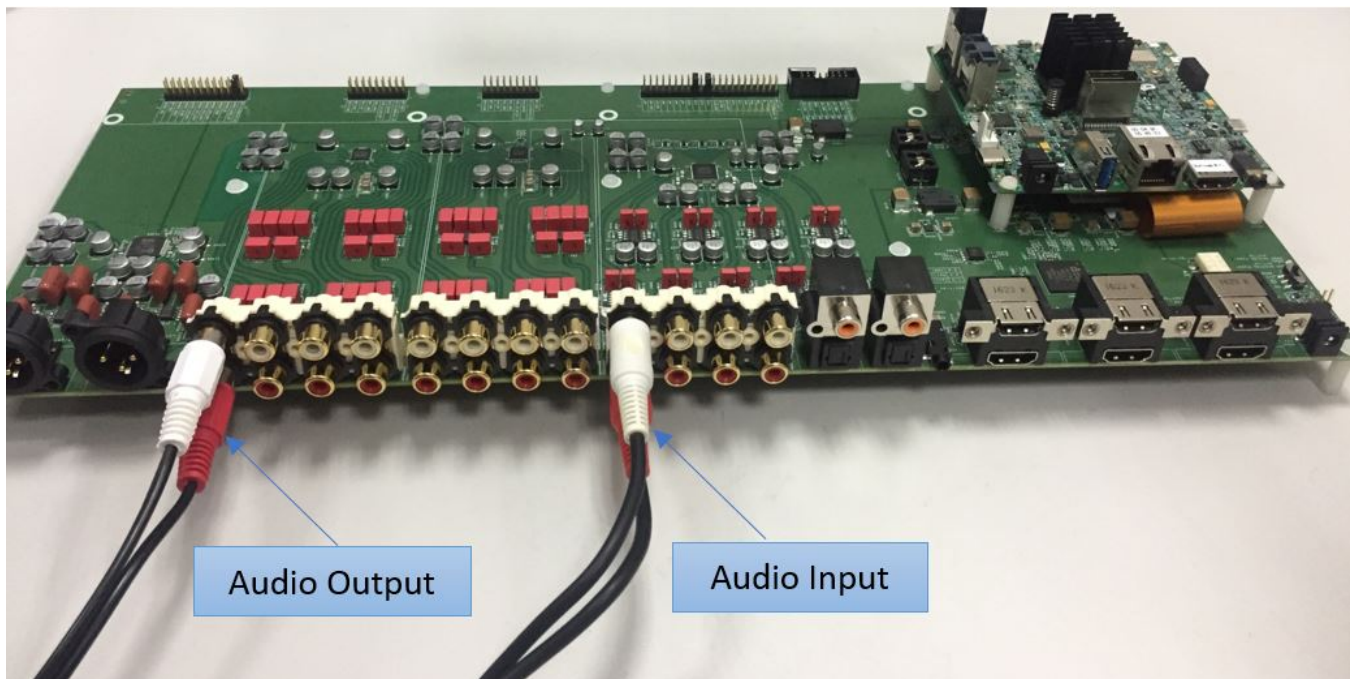


Figure 7. i.MX 8M Quad EVK with audio board



Figure 8. i.MX 8M Mini SAS cable with DSI-to-HDMI adapter





Figure 9. i.MX MIPI panel

## 4.2 Board images

The table below describes the location in the board partitions of the software images in android\_o8.1.0\_1.5.1\_8mm-beta\_image\_8mqevk.tar.gz.

Table 3. Board images

Image name	Download target
u-boot-imx8mq.imx	33 KB offset of SD card.
partition-table.img	0 offset of MMC. If the actually size of the SD card is larger than 13 GB, use the default partition-table.img.
partition-table-7GB.img	0 offset of MMC. If the actually size of the SD card is larger than 7 GB, use this image as partition-table.img.
partition-table-28GB.img	0 offset of MMC. If the actually size of the SD card is larger than 28 GB, use this image as partition-table.img.
boot-imx8mq.img	boot_a and boot_b partitions to support HDMI output.
boot-imx8mq-mipi.img	boot_a and boot_b partitions to support MIPI-DSI to HDMI output.
boot-imx8mq-dual.img	boot_a and boot_b partitions to support HDMI and MIPI-DSI to HDMI dual output.

Table continues on the next page...

**Table 3. Board images (continued)**

boot-imx8mq-mipi-panel.img	boot_a and boot_b partitions to support MIPI panel output.
boot-imx8mq-dsd.img	boot_a and boot_b partitions to support HDMI output and Direct Stream Digital (DSD) playback.
vbmeta-imx8mq-dsd.img	vbmeta_a and vbmeta_b partitions to support HDMI output and DSD playback.
vbmeta-imx8mq.img	vbmeta_a and vbmeta_b partitions to support HDMI output.
vbmeta-imx8mq-mipi.img	vbmeta_a and vbmeta_b partitions to support MIPI-DSI to HDMI output.
vbmeta-imx8mq-dual.img	vbmeta_a and vbmeta_b partitions to support HDMI and MIPI-DSI to HDMI dual output.
vbmeta-imx8mq-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support MIPI panel output.
system.img	system_a and system_b partitions.
vendor.img	vendor_a and vendor_b partitions.

## 4.3 Flashing board images

The board images can be flashed to the target board by using the MfgTool. The release package includes MfgTool for i.MX 8M Quad EVK in android\_o8.1.0\_1.5.1\_8mm-beta\_tools.tar.gz. The MfgTool is mfgtools.zip.

### NOTE

The MfgTool only works in Windows OS environment.

Perform the following steps to flash the board images:

1. Unzip the mfgtools.zip file to a selected location. The directory is named MFGTool-Dir in this example.
2. Copy following files from release\_package/android\_o8.1.0\_1.5.1\_8mm-beta\_image\_8mq.tar.gz to your MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/evk directory.
  - /u-boot-imx8mq.img
  - /partition-table.img
  - /boot-imx8mq.img
  - /vbmeta-imx8mq.img
  - /system.img
  - /vendor.img

### NOTE

- Do not replace any other files in the files directory and the OS Firmware directory.
- If the SD card is 8 GB, copy partition-table-7GB.img and rename it to partition-table.img.
- If the SD card is 16 GB, use the default partition-table.img.
- If the SD card is 32 GB, copy partition-table-28GB.img and rename it to partition-table.img.
- To test HDMI output, copy boot-imx8mq.img and vbmeta-imx8mq.img.
- To test HDMI output with DSD playback supported, copy boot-imx8mq-dsd.img and vbmeta-imx8mq-dsd.img, and rename them to boot-imx8mq.img and vbmeta-imx8mq.img.
- To test MIPI-DSI to HDMI output, copy boot-imx8mq-mipi.img and vbmeta-imx8mq-mipi.img, and rename them to boot-imx8mq.img and vbmeta-imx8mq.img.

- To test HDMI and MIPI-DSI to HDMI dual output, copy boot-imx8mq-dual.img and vbmeta-imx8mq-dual.img, and rename them to boot-imx8mq.img and vbmeta-imx8mq.img.
  - To test MIPI panel output, copy boot-imx8mq-mipi-panel.img and vbmeta-imx8mq-mipi-panel.img, and rename them to boot-imx8mq.img and vbmeta-imx8mq.img.
  - To use onboard eMMC as the boot device, use the default partition-table.img.
3. Change the board's SW802 (boot mode) to 01 (from 1 bit to 2 bit) to enter serial download mode.
  4. Power on the board. Use USB cable on the board USB3.0 port, and connect a computer running Windows OS with the board.

**NOTE**

There are two USB ports in i.MX 8M Quad EVK board: USB to UART, USB 3.0. The USB to UART can be referenced as debug UART, and the USB 3.0 can be referenced as USB in the hardware image above. The debug UART can be used to watch the log of the hardware boot processing.

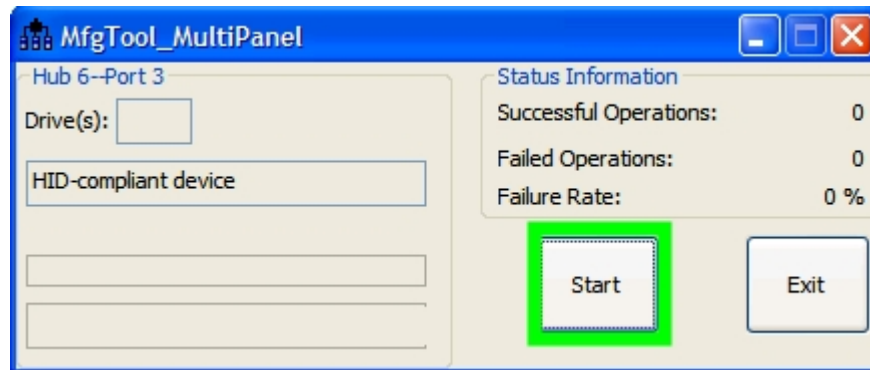
SD Card must be plugged in after the board is powered on.

5. Double-click the \*.vbs file according to the target device as shown in the following table.

**Table 4. MfgTool VBS file**

Target device and boot storage	VBS file
i.MX 8M Quad EVK SD	mfgtool2-android-mx8mq-evk-sd.vbs
i.MX 8M Quad EVK eMMC	mfgtool2-android-mx8mq-evk-emmc.vbs

6. Click Start to start flashing images.



**Figure 10. Starting flash**

The figure below shows flashing in progress, and the status bar shows the flash status. The flash may take one to two minutes depending on the host machine.



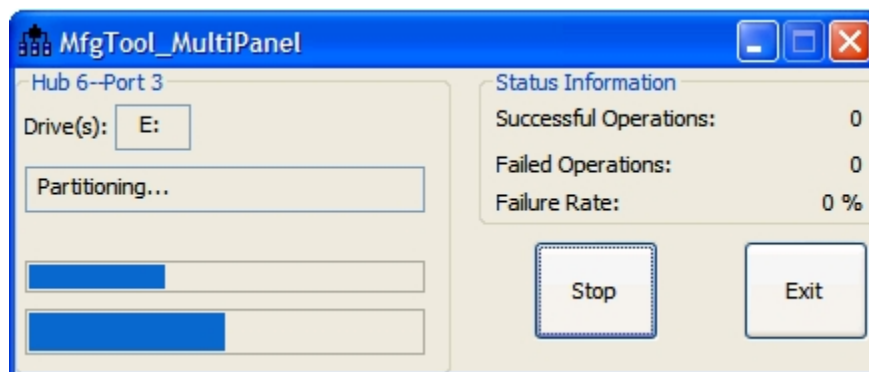


Figure 11. Download status

The figure below shows the tool when the flash is complete.



Figure 12. Download complete

7. Click Stop and disconnect the USB cable.
8. Change SW802 (boot mode) to 10 (from 1 bit to 2 bit). Change SW801 to switch the board back to 1100 (SD boot mode). Change SW801 to switch the board back to 0010 (eMMC boot mode).

## 4.4 Booting

After downloading the images, boot the board by connecting it to the power supply.

### 4.4.1 Booting with single display: HDMI display

In the U-Boot prompt, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200 init=/
init androidboot.gui_resolution=1080p androidboot.console=ttyMXC0 consoleblank=0
androidboot.hardware=freescale androidboot.fbTileSupport=enable cma=1280M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
U-Boot > saveenv
```

With above settings, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

**NOTE**

Boot image supporting DSD playback (boot-imx8mq-dsd.img) also uses Single HDMI Display, and they share the same U-Boot environment variables.

## 4.4.2 Booting with single display: MIPI-DSI to HDMI display

In the U-Boot prompt, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootargs console=ttymx0,115200 earlycon=imxuart,0x30860000,115200 init=/
init androidboot.lcd_density=160 androidboot.console=ttymx0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=mxsfb-drm
firmware_class.path=/vendor/firmware
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

## 4.4.3 Booting with dual displays: HDMI and MIPI-DSI to HDMI displays

In the U-Boot prompt, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootargs console=ttymx0,115200 earlycon=imxuart,0x30860000,115200 init=/
init androidboot.gui_resolution=1080p androidboot.console=ttymx0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware
U-Boot > saveenv
```

With above settings, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

## 4.4.4 Booting with single display: MIPI panel

In the U-Boot prompt, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootargs console=ttymx0,115200 earlycon=imxuart,0x30860000,115200 init=/
init androidboot.console=ttymx0 consoleblank=0 androidboot.hardware=freescale cma=1280M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

## 4.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

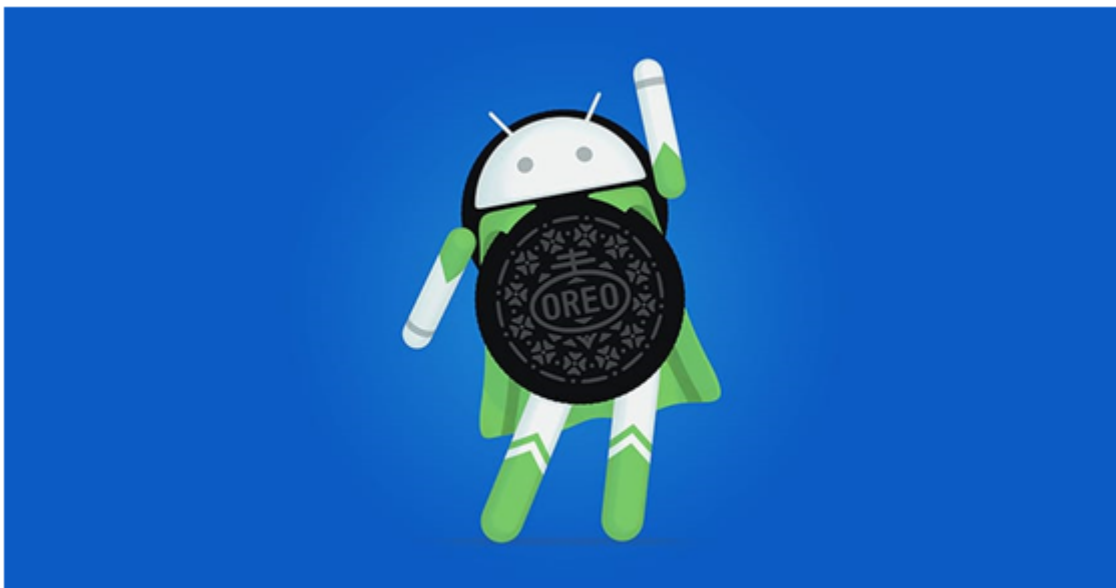


Figure 13. Android Oreo image

## 5 Revision History

Table 5. Revision history

Revision number	Date	Substantive changes
O8.1.0_1.5.0_8MM-alpha	07/2018	Initial release
O8.1.0_1.5.1_8MM_beta	09/2018	i.MX 8M Mini Beta release

**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro,  $\mu$ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018 NXP B.V.

Document Number AQSUG  
Revision 08.1.0\_1.5.1\_8MM-beta, 09/2018

